IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re PATENT application of:

Appellant(s):	Matthew Crewe
Serial No:	10/630,020
Filing Date:	July 30, 2003
Title:	FLEXIBLE INTEGRATION OF SOFTWARE APPLICATIONS IN A NETWORK ENVIRONMENT
Examiner:	Brian P. Whipple
Art Unit:	2152
Docket No.	DYOUP0253US

APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

The undersigned submits this brief for the Board's consideration of the appeal of the Examiner's decision, mailed June 23, 2008, finally rejecting claims 1–19 in the above-identified application.

The fee for filing an appeal brief is being paid herewith. In the event an additional fee is necessary, the Commissioner is authorized to charge any additional fee which may be required to Deposit Account No. 18-0988 under the above shown docket number.

I. Real Party in Interest

The real party in interest in the present appeal is Barcoview MIS Edinburgh, the assignee of record, which is a UK Branch of Barco NV.

II. Related Appeals and Interferences

Neither appellant nor appellant's legal representative are aware of any appeals or interferences which will directly affect, which will be directly affected by, or which will have a bearing on the Board's decision in the pending appeal.

III. Status of Claims

Claims 1–19 are rejected. The claims on appeal are claims 1–19. A correct copy of these claims is reproduced in the Claims Appendix.

IV. Status of Amendments

An Amendment was filed on September 23, 2008, subsequent to issuance of the Final Office Action dated June 23, 2008, from which this appeal is taken. The Amendment addressed the rejection of claims 1–13 under 35 U.S.C. § 101 and the rejection of claims 1–5 under 35 U.S.C. § 112. According to the Advisory Action dated October 2, 2008, the Examiner entered the Amendment and withdrew the rejection of claims 1–13 under 35 U.S.C. § 101 and the rejection of claims 1–5 under 35 U.S.C. § 112.

V. Summary of Claimed Subject Matter

The following is a concise explanation of the subject matter defined in each of the independent claims involved in the appeal and refers to the specification by page and line number, and to the drawing, if any, with reference characters.

Claim 1

Independent claim 1 recites a software product containing a medical-imaging

visualization application A ([12/11-20], Figure 2). As claimed, the software product comprises computer executable instructions embodied on a computer readable medium ([10/21-25]) that are configured when executed to function as a model component M in a model-view-controller software architecture ([12/20-29], Figure 2). The claimed software product also has an interface API ([13/10-17], Figure 2) having a set of user interface control parameters ([14/13-15/10]) and a set of data handling parameters ([15/11-17/14]). As claimed, the sets of parameters are chosen to allow flexible integration of the visualization application into a proprietary Picture Archiving and Communications Systems (PACS) network 2 ([11/2-12/3], Figure 1).

Claim 6

Independent claim 6 recites a PACS network 2 including a logic device 16 for executing instructions ([10/14-25], Figure 1) of a software component containing a medical-imaging visualization application A ([12/11-20] and Figure 2). The claimed software component is configured to function as a model component M in a model-view-controller software architecture ([12/20-29], Figure 2]) and has an interface API ([13/10-17], Figure 2) having a set of user interface control parameters and a set of data handling parameters ([14/13-17/14]). As claimed, the sets of parameters are chosen to allow flexible integration of the visualization application into the PACS network 2 ([11/2-12/3], Figure 1).

Claim 14

Independent claim 14 recites a method of offering a medical-imaging data visualization application to a PACS network integrator ([20/27–28]). As claimed, the method comprises providing a first version of the application contained in a high-level

software component ([20/28–30]) and providing a second version of the application contained in a plurality of lower-level software components ([20/30-21/9]). The claimed method further comprises allowing the integrator to decide between the use of different versions for integrating the application into a PACS network ([21/10-27]).

VI. Grounds of Rejection to Be Reviewed on Appeal

Claims 1–19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,875,327 (herein "Brandt") in view of U.S. Patent No. 6,574,629 B1 (herein "Cooke"). The rejections of each of claims 1–19 are presented for review.

VII. Argument

Rejection under 35 U.S.C. § 103(a) over Brandt in view of Cooke

The Examiner rejects claims 1–19 as being obvious in light of Brandt and Cooke. The rejections advanced by the Examiner are improper and should be reversed for at least the following reasons.

As an aid to understanding the differences between the prior art disclosures and the subject matter of the claims, the following summary of conventional medical-imaging software applications and Picture Archiving and Communications Systems (PACS) networks is provided.

Traditional medical-imaging software applications were once operated as stand-alone applications or loosely integrated independent applications. In an effort to standardize medical-imaging software applications and associated networks, the PACS network architecture was developed to allow more simplistic networking of devices, such as imaging modalities (e.g., CT scanners), data storage devices, and computers.

However, since medical-imaging software applications are independently provided by a variety of providers, each may have a different user interface. As a result, the overall PACS network has a non-uniform appearance and requires network users to become familiar with many different interfaces.

A conventional solution to the above issue allows PACS network providers to provide a preferred unitary user interface with the functionality of the individual software applications by licensing granular versions of the applications. However, the granular multi-component approach requires the skills of a highly skilled technical programmer to integrate the code of the different applications into the base code of the PACS network. The programmer must also have an in-depth understanding of the underlying medical-imaging technology associated with the software application. Thus, the integration requires significant engineering time and expense, and may be prone to coding errors not present in the well-tested stand-alone versions of the software applications.

The foregoing is believed to be the conventional manner of integrating medical-imaging software applications into a PACS network.

The claimed invention seeks to simplify and accelerate the integration of a medical-imaging software application into a PACS network by providing a software product and methodology that can be used by a typical programmer without special knowledge of the application or associated medical-imaging technology. In particular, the claimed invention provides a medical-imaging visualization application within a software product, where the software product is configured to function as a model component in a model-view-controller software architecture.

By definition, a model-view-controller software architecture includes a model

component, a view component, and a controller component. The model component defines the fundamental aspects of the functionality of an application, while the view component manages the graphical and/or textual output, and the controller interprets user inputs in order to direct the model and/or view components to respond appropriately. In the claimed context, the application provider supplies the model component as a self-contained software product, while the PACS network provider provides customized view and controller components that are consistent with the "look and feel" of the PACS network.

In order to assure that the model component of the application is able to communicate with the view and controller components of the PACS network, the software product further includes an interface having appropriate data handling and user interface control parameters that allow flexible integration of the software application into the PACS network, as claimed. For example, the claimed interface may be designed to map the complicated technical functions of the software application to standard graphical user interface building blocks that may be used by the PACS network provider to design the corresponding view and controller components. As a result, the PACS network provider is able to provide a unitary user interface for accessing the functionality of a software application without having to understand the fundamental operation of the application.

Brandt has nothing to do with the problem addressed by the instant application and has much less to do with the solution invented by the appellant. Unlike the claimed invention, Brandt solves a different problem of configuring computer workstations in a network such that a user-selected configuration may be saved and applied to any

workstation on which the user is presently working (col. 2, lines 7–11). In Brandt, this problem is solved by saving a user's preference file on a file server and using the configuration parameters included in the preference file to configure the workstation on which the user is currently working (col. 4, lines 15–25). A second aspect of the Brandt solution includes resolving conflicts between different preferences established by the user, an administrator and/or manufacturer by managing the preference files according to a multi-level hierarchy of precedence (col. 9, lines 5–25, Figure 2). No portion of Brandt has been found to teach or suggest a model-view-controller software architecture, let alone a software product that is configured to function as a model component and has an interface with parameters for allowing integration of the application contained in the software product into a PACS network, as claimed.

In Section 8 of the Final Office Action, the Examiner incorrectly equates the preference files of Brandt with the "model component" of the claimed invention, the user's level of operation in the Brandt network with the "view component" included in the claimed PACS network, and the ability to control the hierarchical order of preference files in Brandt with the "controller component" also included in the PACS network. Appellant respectfully points out that the term "model-view-controller software architecture" has specific meaning in the art of software engineering, as described above, and that one of ordinary skill in the art would not interpret the Brandt disclosure as describing the software architecture, or even the features of such architecture.

In particular, while the claims recite a software product containing a medical-imaging visualization application that includes computer executable instructions that when executed are configured to function as a model component, the preference files in

Brandt embody none of these claimed features. The Brandt preference files do not contain a visualization application, or any other type of application that includes computer executable instructions that are configured to function as a model component, as set forth in the claims. Instead, the preference files of Brandt merely define preferences for configuration parameters such as screen color, mouse operation, and the like (col. 5, lines 45–48). In contrast, a model component defines the fundamental functionality of an application. Hence, the preference files of Brandt cannot be seen as being operable, or configured, to function as a model component in a model-view-controller software architecture.

In Section 9 of the Final Office Action, the Examiner incorrectly equates the configuration parameters included in the Brandt preference files with the data handling parameters and user interface parameters of the interface included in the claimed software product. As explained above, typically, different medical-imaging software applications have different user interfaces, none of which may conform to the user interface of the PACS network, thereby giving a non-uniform appearance to the overall network. According to the claimed invention, in order to allow a PACS network provider to easily provide the functionality of different software applications to a user through a unitary user interface, the claimed interface includes parameters that allow the model component to interact with the user via the PACS network and to allow the user to access the functionality of the model component via the PACS network (Spec., pg. 13, lines 10–17). That is, the claimed parameters are chosen to allow integration of the visualization application into a PACS network. Nothing in Brandt has been found to disclose the claimed subject matter.

If the Examiner's analogy were correct, the Brandt configuration parameters would be designed as part of an interface that allows the preference files (purportedly teaching the claimed software product) to be integrated into a network. This is not the case. The configuration parameters of Brandt merely represent hardware or software parameters (e.g., screen color or mouse operation) that may be configured according to personal preferences. Unlike the claimed invention, neither the Brandt configuration parameters nor the preference files are concerned with allowing an application to be integrated into a network such that the application may be presented to a user with the unitary "look and feel" of the PACS network. In fact, Brandt discloses the opposite of achieving a common user interface. The problem addressed and solved by Brandt is personalizing the settings of an individual workstation according to the different preferences of the associated user, manufacturer, and/or administrator, while managing any conflicts between the different preferences. Accordingly, the preference files and configuration parameters in Brandt are distinct from a software product having an interface that has user interface control parameters and data handling parameters, where the parameters are chosen to allow flexible integration of the application into a PACS network, as claimed.

Noting that Brandt does not teach a PACS network or a medical-imaging visualization application, the Examiner proposes modifying the teachings of Brandt with the teachings of a medical-imaging visualization application and a PACS network in Cooke. Even if the proposed modification were made, the claimed invention would not result. While Cooke mentions a PACS network with applications for displaying medical images, any reasonable combination of Brandt and Cooke would, at most, provide a

PACS network that includes a server for storing preference files that may be used to configure a workstation according to the personal preferences of a user logged onto that workstation. Like Brandt, Cooke does not disclose a software product containing a medical-imaging visualization application that includes computer executable instructions that are configured to function as a model component in a model-view-controller software architecture and has an interface having a set of parameters that are chosen to allow flexible integration of the visualization application into a PACS network, as claimed.

Claims 1, 2, and 5

In view of the foregoing discussion, it should be evident that even if Brandt were modified by Cooke, the proposed modification would not result in the subject matter recited in claim 1. In particular, neither of the references, alone or in combination, discloses a software product containing a medical-imaging visualization application, where the software product comprises computer executable instructions embodied on a computer readable medium that are configured when executed to function as a model component in a model-view-controller software architecture, and has an interface having a set of user interface control parameters and a set of data handling parameters, the sets of parameters being chosen to allow flexible integration of the visualization application into a PACS network, as recited in claim 1.

Claims 2 and 5 depend from claim 1, and therefore, recite novel and inventive subject matter for at least the reasons discussed above.

Claim 3

Claim 3 depends from claim 1 and further calls for the software product being a sub-component of a pre-existing data visualization application. Contrary to the Examiner's contention in Section 28 of the Final Office Action, the combination of Brandt and Cooke does not disclose this feature. The portions of Brandt referenced by the Examiner (i.e. Figure 2; col. 9, lines 9–14) simply disclose a hierarchical scheme of precedence used to manage conflicting preferences for configuration parameters. The portions of Cooke referenced by the Examiner (i.e. Abstract; col. 7, lines 49–54) disclose an application for displaying medical images in a PACS network. Neither Brandt nor Cooke discloses a software product that is a sub-component of a pre-existing data visualization application, where the application is integrated into a PACS network, as claimed. Therefore, the combination of the hierarchical scheme in Brandt with the applications for displaying medical images in Cooke does not result in the claimed invention.

Claim 4

Claim 4 depends from claim 3 and further calls for the software product including a software wrapper, where the software wrapper is configured to map the sets of parameters of the interface to parameters appropriate for the sub-component. Contrary to the Examiner's contention in Section 29 of the Final Office Action, Brandt does not discuss the claimed subject matter. The passages of Brandt referenced by the Examiner (i.e. col. 9, lines 9–25) simply disclose a hierarchical scheme of precedence used to manage conflicting preferences for configuration parameters. As described above, no portion of Brandt describes the claimed software product or an interface having a set of parameters chosen to allow integration of an application in to a PACS

network. Therefore, Brandt cannot be found to disclose a software product that includes a software wrapper as claimed.

Moreover, no portion of Brandt discloses the features recited with specificity in claim 4. The claimed software wrapper is configured to map the sets of parameters of the interface to parameters appropriate for the sub-component in order to simplify the integration of the application into a PACS network. For example, as explained in the specification, the software wrapper may be designed to allow access to the rotation function of an application without requiring the provider to become familiar with the concept of defining elements in a rotation matrix operator (pg. 6, lines 6–20). Brandt simply discloses the use of configuration parameters to configure a workstation according to the preferences of select parties. The Brandt configuration parameters are not mapped to parameters appropriate for a sub-component of a pre-existing application, as claimed. Hence, Brandt does not disclose the claimed subject matter.

Claims 6 and 7

Claim 6 recites a PACS network that includes a logic device for executing instructions of a software component containing a medical-imaging visualization application, where the software component is configured to function as a model component in a model-view-controller software architecture and has a set of user interface control parameters and a set of data handling parameters, the sets of parameters being chosen to allow flexible integration of the visualization application into the PACS network. Contrary to the Examiner's contention in Section 26 of the Office Action, neither of the references, alone or in combination, discloses the subject matter of claim 6.

For at least the reasons discussed above, nothing in Brandt describes a software product containing an application, where the software product is configured to function as a model component in a model-view-controller software architecture. In addition, no portion of Brandt describes a software product that has an interface having a set of parameters that are chosen to allow flexible integration of the visualization application into a PACS network. In fact, neither Brandt nor Cooke has anything to do with the integration of an application into a network. Accordingly, the combination of Brandt and Cooke has not been found to disclose the features recited in claim 6.

Claim 7 depends from claim 6 and therefore, recites novel and inventive subject matter for at least the reasons discussed above.

Claims 8–12

Claim 8 depends from claim 6 and further calls for the PACS network including a specific glue bridge software component that is configured to accommodate non-standard aspects of the PACS network. Contrary to the Examiner's contention in Section 31 of the Final Office Action, Brandt does not disclose this feature. The portions of Brandt referenced by the Examiner (i.e. col. 4, lines 43–55) simply describe the ability of the Brandt system to overwrite the preferences of an administrator with the preferences of an individual user. One of ordinary skill in the art would not view these or any other portions of Brandt as describing the claimed specific glue bridge software component.

As explained in the specification, a PACS network may have non-standard aspects where a PACS network provider decides to deviate from standard software practices (pg. 18, lines 1–26). This makes it difficult for an application provider to

supply a generic model component that can be integrated into all PACS networks in accordance with the claimed approach. To address this problem, the claimed PACS network may include a specific glue bridge that comprises, for example, logic designed to allow the otherwise incompatible model component to be integrated into a particular PACS network despite the non-standard aspects of that network. Brandt has nothing to do with the use of a specific glue bridge software component, or its equivalent, as recited in claim 8.

Claims 9 through 12 depend from claim 8 and therefore, recite novel and inventive subject matter for at least the reasons discussed above.

Claim 13

Claim 13 depends from claim 6 and further calls for the PACS network including a dispatcher software component that is configured to link multiple software components corresponding to multiple software applications to the PACS network via a common interface. Contrary to the Examiner's contention in Section 36 of the Final Office Action, the combination of Brandt and Cooke does not disclose this feature. Rather, the portions of Brandt referenced by the Examiner simply show a network system comprised of a plurality of workstations (Figure 1) and the interaction between a preference manager and a plurality of hierarchical preference files to create a resultant set of references (Figure 2). And the portions of Cooke referenced by the Examiner simply show a typical PACS cluster that includes several modules and workstations. Neither Brandt nor Cooke describes linking multiple software components to the PACS network via a common interface, where the multiple software components correspond to multiple software applications, as claimed.

As explained in the specification, in accordance with the claimed invention, an exemplary way to integrate multiple model components into the PACS network involves providing a dispatcher that serves as a common interface between the different model components and the PACS network. For example, the dispatcher may be responsible for routing instructions from the controller component of the PACS network to the appropriate model component and for passing display information from the relevant model component to the view component of the PACS network (pg. 19, lines 4–23). Since neither of the references discloses the use of a software component as a model component in a model-view-controller software architecture in order to integrate an application into a PACS network, as discussed above, the references cannot be found to disclose a dispatcher that is configured to link multiple software components corresponding to multiple software applications to the PACS network via a common interface, as recited in claim 13.

Claims 14 and 17

Claim 14 recites a method of offering a medical-imaging data visualization application to a PACS network integrator, where the method comprises providing a first version of the application contained in a high-level software component, providing a second version of the application contained in a plurality of lower-level software components, and allowing the integrator to decide between use of the different versions for integrating the application into a PACS network. Contrary to the Examiner's contention in Section 20 of the Final Office Action, the combination of Brandt and Cooke does not disclose the subject matter of claim 14.

The Examiner equates the hierarchical levels used by the preference manager to arrange preference files in Brandt with the recitation of "a first-version of the application contained in a high-level software component" and "a second version of the application contained in a plurality of lower-level software components" in claim 14. To support this contention, the Examiner refers to the following portion of Brandt:

> Again, the merging or coalescing of preference files will typically utilize some hierarchical scheme of precedence where information from preference files located in higher levels of the hierarchy take precedence over information from preference files in lower levels of hierarchy (see FIG. 2). However, it is recognized that under different circumstances, configuration parameters from lower level preference files may take precedence over configuration parameters from higher level preference files.

(col. 9, lines 9–18).

Upon reading the above cited text, one of ordinary skill in the art would quickly recognize that Brandt has nothing to do with software components containing applications, let alone software components containing data visualization applications, as set forth in claim 14. Rather, the cited portions demonstrate that Brandt is only concerned with ranking preference files to determine which preferences should be used to configure the parameters of a workstation, and that the term "level" in the Brandt context refers to the hierarchical arrangement of preference files, where, for example, an administrator's preferences may be ranked higher than those of a user.

In contrast, the terms "high-level software component" and "lower-level software component" in claim 14 take their usual conventional meaning. In the context of software, "level" refers to the software code's position within the

functionality of the computer system. For example, "low level" refers to the programming and processing that deal with the very fundamental inner operation of the computer and therefore, are more readily transferable between computer systems. "High level," on the other hand, refers to the operations and interactions that are experienced by and evident to the user, and therefore, are more specific to an individual system or application. Hence, the hierarchical levels described in Brandt have nothing to do with high-level and lower-level software components, as recited in claim 14. Cooke offers little, if anything, in this context.

Accordingly, based on the foregoing, the combination of Brandt and Cooke has not been found to disclose at least the following features of claim 14: providing a first version of the application contained in a high-level software component; providing a second version of the application contained in a plurality of lower-level software components, and allowing the integrator to decide between use of the different versions for integrating the application into a PACS network.

Claim 17 depends from claim 14 and therefore, recites novel and inventive subject matter for at least the reasons above.

Claims 15 and 16

Claim 15 depends from claim 14 and further calls for the high-level software component being configured to function as a model component in a model-view-controller software architecture, and having an interface having a set of user interface control parameters and a set of data handling parameters. Contrary to the Examiner's

contention in Section 21 of the Final Office Action, the proposed combination of Brandt and Cooke does not disclose the features of claim 15. For at least the reasons discussed above, neither Brandt nor Cooke describes a software component configured to function as a model component in a model-view-controller software architecture, as claimed. Nor do the cited references disclose a software component having an interface having a set of user interface control parameters and a set of data handling parameters, as claimed. Accordingly, the combination of Brandt and Cooke has not been found to disclose the subject matter of claim 15.

Claim 16 depends from claim 15 and therefore, recites novel and inventive subject matter for at least the reasons above.

Claim 18

Claim 18 depends from claim 14 and further calls for providing a third version of the application contained in a plurality of intermediate-level software components. Contrary to the Examiner's contention in Section 24 of the Final Office Action, Brandt does not disclose the features of claim 18. As discussed above, Brandt does not disclose the use of software components containing applications of any kind. Nor does Brandt disclose different levels of software components according to the meaning of "level" known in the art. Therefore, Brandt cannot be found to disclose a third version of the application contained in a plurality of intermediate-level software components, as recited in claim 18.

Claim 19

Claim 19 depends from claim 18 and further calls for providing at least a fourth version of the application contained in a plurality of software components of a different

level. Contrary to the Examiner's contention in Section 25 of the Final Office Action, Brandt does not disclose the features of claim 19. As discussed above, Brandt does not disclose the use of software components containing applications of any kind. Nor does Brandt disclose different levels of software components according to the meaning of "level" known in the art. Accordingly, Brandt cannot be found to disclose a fourth version of the application contained in a plurality of software components of a different level, as recited in claim 19.

For at least the above reasons, the rejection of claims 1–19 is improper.

VIII. Conclusion

In view of the foregoing, it is respectfully submitted that the claims are patentable over the applied art and that the rejections advanced by the Examiner should be reversed.

Respectfully submitted,

RENNER, OTTO, BOISSELLE & SKLAR, L.L.P.

By: /Timothy E. Manning/
Timothy E. Manning, Reg. No. 48,964

1621 Euclid Avenue
Nineteenth Floor
Cleveland, Ohio 44115
216-621-1113

Claims Appendix

What is claimed is:

1. A software product containing a medical-imaging visualization application, the software product comprising computer executable instructions embodied on a computer readable medium that are configured when executed to function as a model component in a model-view-controller software architecture, and having an interface having a set of user interface control parameters and a set of data handling parameters, the sets of parameters being chosen to allow flexible integration of the visualization application into a proprietary Picture Archiving and Communications Systems (PACS) network.

2. A software product according to claim 1, wherein the data handling parameters are Digital Imaging and Communications in Medicine (DICOM) format data handling parameters.

3. A software product according to claim 1, wherein the software product is a sub-component of a pre-existing data visualization application.

4. A software product according to claim 3, wherein the software product includes a software wrapper, the software wrapper being configured to map the sets of parameters of the interface to parameters appropriate for the sub-component.

5. A software product according to claim 1, wherein the user interface control parameters include any of: two-dimensional (2D) tool parameters, three-dimensional (3D) tool parameters, sculpting parameters, display decoration parameters, preset parameters, region of interest select parameters, volume rendering parameters and image display parameters.

6. A PACS network including a logic device for executing instructions of a software component containing a medical-imaging visualization application, the software component configured to function as a model component in a model-view-controller software architecture, and having an interface having a set of user interface control parameters and a set of data handling parameters, the sets of parameters being chosen to allow flexible integration of the visualization application into the PACS network.

7. A PACS network according to claim 6, wherein the data handling parameters are DICOM format data handling parameters.

8. A PACS network according to claim 6, the PACS network including a specific glue bridge software component, the specific glue bridge being configured to accommodate non-standard aspects of the PACS network.

9. A PACS network according to claim 8, wherein the non-standard aspects of the PACS network include a non-standard data format.

10. A PACS network according to claim 9, wherein the non-standard data format is a compressed data format.

11. A PACS network according to claim 8, wherein the non-standard aspects of the PACS network include non-standard data handling.

12. A PACS network according to claim 11, wherein the non-standard data handling relates to proprietary grouping of data.

13. A PACS network according to claim 6, the PACS network including a dispatcher software component, the dispatcher being configured to link multiple software components corresponding to multiple software applications to the PACS network via a common interface.

14. A method of offering a medical-imaging data visualization application to a PACS network integrator, the method comprising:

providing a first version of the application contained in a high-level software component;

providing a second version of the application contained in a plurality of lower-level software components; and

allowing the integrator to decide between use of the different versions for integrating the application into a PACS network.

15. A method according to claim 14, wherein the high-level software component is configured to function as a model component in a model-view-controller software architecture, and has an interface having a set of user interface control parameters and a set of data handling parameters.

16. A method according to claim 15, wherein the data handling parameters are DICOM format data handling parameters.

17. A method according to claim 14, wherein at least a subset of the lower-level software components relate to underlying technical functions of the application.

18. A method according to claim 14, further comprising:

providing a third version of the application contained in a plurality of intermediate-level software components.

19. A method according to claim 18, further comprising:

providing at least a fourth version of the application contained in a plurality of software components of a different level.

Evidence Appendix

None.

Related Proceedings Appendix

None.